Amendment to the Claims:

This listing of claims will replace all versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a multi-slice network processor system comprising a plurality of processing slice modules, each module processing and storing a slice of packet data, a method for processing a packet in packet slices for transfer over a network interface comprising:

prepending a system header to the packet, the system header providing information for use by the multi-slice system;

assigning a packet identifier to the packet;

segmenting data of the packet into cells, the data including both header and body data for the packet;

generating cell descriptive information for each cell, the cell descriptive information including the packet identifier, and a packet position indicator indicating an order position of data of the cell with respect to the packet;

delivering one or more cells of the packet to one or more processing slice modules based upon load balancing criteria;

storing one or more cells in a buffer in the packet slice; and

generating a buffer correlation data structure correlating the buffer of the packet slice to the packet, wherein the buffer correlation data structure is a linked list of buffer identifiers;

wherein the buffer correlation data structure comprises data representative of cells within a slice.

2. (Original) The method of claim 1 wherein load balancing criteria includes that no load balancing is in effect.

3. (Original) The method of claim 1 wherein the packet identifier is a sequence number representing an order of the packet in a communications flow and further

comprising assigning a communications flow indicator to the cell descriptive information of each cell of the packet.

4. (Original) The method of claim 1 wherein the cell descriptive information further comprises a slice position indicator indicating an order position of the data of the cell with respect to a slice of data of the packet.

5. (Original) The method of claim 3 further comprising delivering body data of the packet to one or more of the processing slices ahead of the header data of the packet.

6. (Original) The method of claim 4 further comprising:

performing lookup functions for each slice of data;

determining a size of data change in header data; and

communicating the size of data change to a queue manager via an indicator in the system header.

7. (Canceled)

8. (Previously Presented) The method of claim 1 further comprising:
generating a slice correlation data structure for the packet based upon a packet reference pointing to the buffer of the packet slice including the first cell of the packet, and the packet identifier in each cell's descriptive information.

9. (Canceled)

10. (Previously Presented) The method of claim 1 further comprising:

generating a slice correlation data structure for the packet including a packet reference pointing to the buffer of the packet slice including the first cell of the packet, and a respective buffer indicator for the buffer in each packet slice storing the first cell in the slice for the packet; and

entering the slice correlation data structure as a single queue entry into a queue.

11. (Previously Presented) The method of claim 1 wherein the network interface is a switch fabric and further comprising determining a destination slice across the switch fabric for each packet slice in accordance with load balancing criteria.

12. (Original) The method of claim 11 further comprising:

for a received packet from the switch fabric, storing each cell of each packet slice of the received packet, each cell including descriptive information, in the processing slice identified in a destination slice indicator of the descriptive information.

13. (Original) The method of claim 12 further comprising sending an enqueue message for each packet slice identifying a storage location of the first cell of the slice.

14. (Original) The method of claim 13 further comprising:

generating a slice correlation data structure for the packet based upon the storage location of the first cell of each slice of the packet, and the packet identifier in each cell's descriptive information;

responsive to the size of data having been changed as indicated in the indicator in the system header, determining packet size adjustment; and

entering the slice correlation data structure as a single queue entry into a queue.

15. (Original) The method of claim 13 further comprising:

upon initiation of retrieval of the packet, generating a new packet identifier for the packet;

sending a dequeue message for each slice of the packet;

correlating each cell of the packet into packet form based on cell descriptive information including the packet position indicator and the slice position indicator; and

ordering the packet for transmission to an attached network based on the new packet identifier.

16. (Currently Amended) A multi-slice network processor system comprising:

a plurality of parallel processing slices, each processing slice comprising a lookup processing module and access to a storage sub-system, the storage sub-system including a memory, the memory storing at least one group of cells of a packet in a buffer; and a buffer manager, the buffer manager maintaining a buffer correlation data structure for correlating one or more buffers to the packet, the buffer correlation data structure comprising data representative of cells with a slice, wherein the buffer correlation data structure is a linked list of buffer identifiers, the buffer correlation data structure being stored in the memory;

a network data distribution and aggregation module for segmenting a packet received from a network into one or more packet slices, the network data distribution and aggregation module having a communication interface to each of the processing slices for communicating each packet slice;

each of the plurality of slices having a channel communication interface with the network interface over which each packet slice is directed to a destination processing slice across the network interface; and

a queuing module having an enqueuing communication interface and a de-queuing communication interface with each of the processing slices, the queuing module controlling the enqueuing and dequeuing of each of the packet slices, and determining the destination processing slice based on load balancing criteria.

17. (Original) The system of claim 16 wherein load balancing criteria includes no load balancing.

18. (Original) The system of claim 16 wherein the network interface is a switch fabric, and wherein each channel communication interface comprises a port connection with the switch fabric.

19. (Canceled)

20. (Original) The system of claim 16 wherein the queuing module includes a queuing memory space, the queuing module maintaining a slice correlation data structure for correlating

one or more slices of the same packet slice in a single queue entry, the slice correlation data structure being stored in the queuing memory space.

21. (Previously Presented) The system of claim 16 wherein the buffer manager comprises an ingress buffer manager including an ingress buffer memory space for each processing slice, the ingress buffer memory space for storing cells received from the respective processing slice, and an egress buffer memory space for each processing slice, the egress buffer memory space for storing cells received from the switch fabric for each respective processing slice.

22. (New) In a multi-slice network processor system comprising a plurality of processing slice modules, each module processing and storing a slice of packet data, a method for processing a packet in packet slices for transfer over a network interface comprising:

prepending a system header to the packet, the system header providing information for use by the multi-slice system;

assigning a packet identifier to the packet;

segmenting data of the packet into cells, the data including both header and body data for the packet;

generating cell descriptive information for each cell, the cell descriptive information including the packet identifier, and a packet position indicator indicating an order position of data of the cell with respect to the packet;

delivering one or more cells of the packet to one or more processing slice modules based upon load balancing criteria;

storing one or more cells in a buffer in the packet slice;

generating a buffer correlation data structure correlating the buffer of the packet slice to the packet, wherein the buffer correlation data structure is a linked list of buffer identifiers;

maintaining an independent set of upper bits of a sequence number for each communication flow; and

responsive to detecting one of the processing slices delivering a sequence number that is smaller in value than an immediately preceding sequence value for the same slice, incrementing

the independent set of upper bits for the respective communication flow, concatenating the set of upper bits with a set of bits of the sequence number into an index, indexing into a re-sequencing buffer space of sufficient depth to cover a slice-to-slice skew case based on the index, and resequencing the packet into its sequence order position.

23. (New) The method of claim 22 wherein load balancing criteria includes that no load balancing is in effect.

24. (New) The method of claim 22 wherein the packet identifier is a sequence number representing an order of the packet in a communications flow and further comprising assigning a communications flow indicator to the cell descriptive information of each cell of the packet.

25. (New) The method of claim 22 wherein the cell descriptive information further comprises a slice position indicator indicating an order position of the data of the cell with respect to a slice of data of the packet.

26. (New) The method of claim 25 further comprising delivering body data of the packet to one or more of the processing slices ahead of the header data of the packet.

27. (New) The method of claim 26 further comprising:
performing lookup functions for each slice of data;
determining a size of data change in header data; and
communicating the size of data change to a queue manager via an indicator in the system header.

28. (New) The method of claim 22 further comprising generating a slice correlation data structure for the packet including a packet reference pointing to the buffer of the packet slice including the first cell of the packet, and a respective buffer indicator for the buffer in each packet slice storing the first cell in the slice for the packet.

29. (New) The method of claim 28 further comprising entering the slice correlation data structure as a single queue entry into a queue.

30. (New) A multi-slice network processor system comprising:

a plurality of parallel processing slices, each processing slice comprising a lookup processing module and access to a storage sub-system, the storage sub-system including a memory, the memory storing at least one group of cells of a packet in a buffer; and a buffer manager, the buffer manager maintaining a buffer correlation data structure for correlating one or more buffers to the packet, wherein the buffer correlation data structure is a linked list of buffer identifiers, the buffer correlation data structure being stored in the memory;

a network data distribution and aggregation module for segmenting a packet received from a network into one or more packet slices, the network data distribution and aggregation module having a communication interface to each of the processing slices for communicating each packet slice;

each of the plurality of slices having a channel communication interface with the network interface over which each packet slice is directed to a destination processing slice across the network interface; and

a queuing module having an enqueuing communication interface and a de-queuing communication interface with each of the processing slices, the queuing module controlling the enqueuing and dequeuing of each of the packet slices, and determining the destination processing slice based on load balancing criteria;

wherein the queuing module is configured to maintain an independent set of upper bits of a sequence number for each communication flow; and

wherein the queuing module is configured to increment the independent set of upper bits for a communication flow, concatenate the set of upper bits with a set of bits of the sequence number into an index, index into a re-sequencing buffer space of sufficient depth to cover a slice-to-slew case based on the index, and re-sequence the packet into its sequence order position responsive to detecting one of the processing slices delivering a sequence number that is smaller in value than an immediately preceding sequence value for the same slice.

31. (New) The system of claim 30 wherein load balancing criteria includes no load balancing.

32. (New) The system of claim 30 wherein the network interface is a switch fabric, and wherein each channel communication interface comprises a port connection with the switch fabric.

33. (New) The system of claim 30 wherein the queuing module includes a queuing memory space, the queuing module maintaining a slice correlation data structure for correlating one or more slices of the same packet slice in a single queue entry, the slice correlation data structure being stored in the queuing memory space.

34. (New) The system of claim 30 wherein the buffer manager comprises an ingress buffer manager including an ingress buffer memory space for each processing slice, the ingress buffer memory space for storing cells received from the respective processing slice, and an egress buffer memory space for each processing slice, the egress buffer memory space for storing cells received from the switch fabric for each respective processing slice.